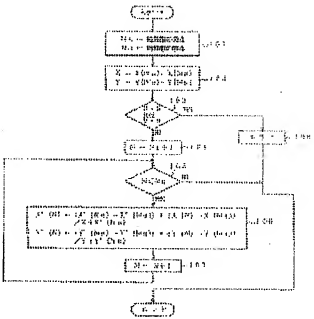
HINTING PROCESSING METHOD

Publication number: JP5323937 (A)
Publication date: 1993-12-07
Inventor(s): SAKAI MINORU
Applicant(s): RICOH KK
Classification:
- international: ***G09G5/24; G06T11/20;*** **G09G5/24; G06T11/20;** (IPC1-7): G09G5/24; G06F15/72
- European:
Application number: JP19920127556 19920520
Priority number(s): JP19920127556 19920520

Also published as:
JP3224142 (B2)

Abstract of **JP 5323937 (A)**

PURPOSE:To hold relative positions of control points at the time of designing and control the contour line by a simple process with a small volume of data. CONSTITUTION:The dot numbers of the contour line are held in Ns and Ne (step 101), and the difference between the (x) coordinates of Ns and Ne is held in X and the difference between the (y) coordinates is held in Y (step 102). Then X(N) is regarded as the (x) coordinate of a dot N and Y(N) is regarded as the (y) coordinate of the dot N; and X'(N) is regarded as the (x) coordinate of the dot N after movement and Y'(N) is regarded as (y) coordinate of the dot N after the movement, and a point right after the starting point is set to N (step 104). Further, N'(N) and T'(N) are calculated until the point N reaches the end point Ns, and N is increased to N+1 and the calculation is repeated (step 105, 106, and 107).; Consequently, points between the dots Ns and Ne are moved so that the initial relative positions are maintained.



(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号

特開平5－323937

(43)公開日　平成５年(1993)12月７日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０９Ｇ　5/24		9061－5G		
Ｇ０６Ｆ　15/72	３５５　Ｕ	9192－5L		

審査請求　未請求　請求項の数11(全 14 頁)

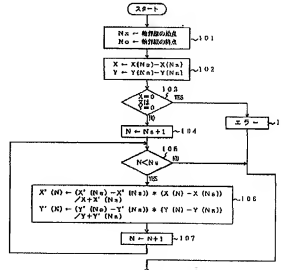
(21)出願番号　特願平4－127556

(22)出願日　平成４年(1992)５月20日

(71)出願人　000006747
株式会社リコー
東京都大田区中馬込１丁目３番６号
(72)発明者　境井　稔
東京都大田区中馬込１丁目３番６号　株式会社リコー内
(74)代理人　弁理士　鈴木　誠　（外１名）

(54)【発明の名称】　ヒンティング処理方法

(57)【要約】
【目的】　デザイン時の輪郭点の相対的な位置を保持し、かつ簡単な処理とより少ないデータ量で輪郭線を制御する
【構成】　輪郭線の始点と終点の点番号をＮｓ，Ｎｅに保持し（ステップ１０１）、ＮｓとＮｅのｘ座標の差をＸに、ｙ座標の差をＹに保持する（ステップ１０２）。Ｘ（Ｎ）を点Ｎのｘ座標、Ｙ（Ｎ）を点Ｎのｙ座標とし、Ｘ’（Ｎ）を点Ｎの移動後のｘ座標、Ｙ’（Ｎ）を点Ｎの移動後のｙ座標とし、始点の次の点をＮに設定する（ステップ１０４）。点Ｎが終点Ｎｓになるまで、Ｘ’（Ｎ）とＹ’（Ｎ）を計算し、ＮをＮ＋１にして繰り返す（ステップ１０５、１０６、１０７）。これにより、点Ｎｓと点Ｎｅに挟まれた点を、最初の相対位置を保つように移動処理する。



1

【特許請求の範囲】

【請求項1】　複数のグリフデータを有し、該各グリフデータは、輪郭点の集合であるアウトライン情報と該アウトライン情報を制御するヒンティング情報によって構成されてなるアウトラインフォントファイルにおける輪郭線のヒンティング処理方法において、輪郭線を構成する所定の輪郭点に対しては、線幅補正、線消え防止および小さずれ防止を含む第1のヒンティング処理を施し、それ以外の輪郭点に対しては、該第1のヒンティング処理された点との相対位置を保つように移動する第2のヒンティング処理を施すことを特徴とするヒンティング処理方法。

【請求項2】　曲線を制御する始点、終点を除く点に対して、前記第2のヒンティング処理を施すことを特徴とする請求項1記載のヒンティング処理方法。

【請求項3】　縦横ステムを構成する輪郭点に対して前記第1のヒンティング処理を施し、それ以外の輪郭点に対して前記第2のヒンティング処理を施すことを特徴とする請求項1記載のヒンティング処理方法。

【請求項4】　縦横ステムを構成する輪郭点と曲線の始点と終点に対して前記第1のヒンティング処理を施すことを特徴とする請求項3記載のヒンティング処理方法。

【請求項5】　曲線中の方向変化を生じる部分に対して前記第1のヒンティング処理を施すことを特徴とする請求項4記載のヒンティング処理方法。

【請求項6】　長方形の対角線上の2点に対して前記第1のヒンティング処理を施し、他の2点に対して前記第2のヒンティング処理を施すことを特徴とする請求項1記載のヒンティング処理方法。

【請求項7】　グリフにおけるウロコの先端点に対して前記第2のヒンティング処理を施すことを特徴とする請求項1記載のヒンティング処理方法。

【請求項8】　本来等しい幅であるべきステムが変倍後においてもその線幅を等しく保つように線幅を補正するヒンティング処理方法において、該線幅と変倍後のサイズは所定の関数関係にあることを特徴とするヒンティング処理方法。

【請求項9】　変倍後のサイズと該サイズに応じた線幅との対応を記憶したテーブルを参照して線幅を補正することを特徴とする請求項8記載のヒンティング処理方法。

【請求項10】　変倍後のサイズに応じて、請求項8記載の関数または請求項9記載のテーブルの何れかを用いて線幅を補正することを特徴とするヒンティング処理方法。

【請求項11】　前記請求項1ないし7記載の処理と前記請求項8ないし10記載の線幅補正処理を組み合わせることを特徴とするヒンティング処理方法。

【発明の詳細な説明】

【0001】

2

【産業上の利用分野】本発明は、アウトラインフォントファイルにおける輪郭線のヒンティング処理方法に関する。

【0002】

【従来の技術】デスクトップパブリッシング、パーソナルコンピュータ、ワードプロセッサ、ディジタルファックス等においては、その装置内に文字フォントデータが用意されている。斯るフォントデータは、通常、ドットパターン方式、アウトライン方式、あるいはそれらの併用方式の何れかの方式によって保持されている。そして、アウトラインフォントを表示、印刷するために、そのフォント表現はドットパターンに変換される。

【0003】従来、アウトラインフォントをドットイメージに変換するラスタライズにおいては、フォントファイルによって輪郭点の座標データと各輪郭点のヒンティングに関する属性データが与えられ、実際のヒンティング処理は、これら座標データと属性データを用いてラスタライザが行っていた。

【0004】ここで、ヒンティング処理とは、アウトラインを美しく整形するための処理をいい、ヒンティングに際して輪郭点を適当な位置に移動することによって行う。従来、このヒンティング処理は、四捨五入や切り捨てなどの丸め処理を行っているために、本来滑らかにつながるようにデザインされた複数の曲線がヒンティング処理後には凸凹になり易いという傾向があった。

【0005】

【発明が解決しようとする課題】上記した点を改善する技術としては、例えば、特開平2－223988号公報に記載の文字処理装置が提案されている。この装置は、文字パターンを分類分けし、分類されたパターンについて条件に合った平滑化処理を施すものであるが、その処理は非常に複雑である。

【0006】ところで、この種の技術分野においては、アウトラインフォントのサイズが大きくなるにしたがって線が細くなって見えるという問題があることが良く知られている。これは、ステムなどの黒部分が太くなるのと同時に、その隙間の白部分も広くなり、この結果、人間の目には白部分のみが強調されて相対的に線が細く見えてしまうと説明される。

【0007】そこで、これを解決した技術として、特公昭63－6874号公報に記載の文字・図形の発生方法があるが、この方法は一つの文字の輪郭部について2つのパターンを文字メモリに格納して処理しているので、処理データ量が通常の2倍近くになるという欠点があった。

【0008】本発明の目的は、デザイン時の輪郭点の相対的な位置を保持し、かつ簡単な処理と、より少ないデータ量で輪郭線を制御するヒンティング処理方法を提供することにある。

【0009】本発明の他の目的は、変倍後のサイズに適

応して線幅を変化させ、サイズに応じて高品質に処理するヒンティング処理方法を提供することにある。

【0010】

【課題を解決するための手段】前記目的を達成するために、請求項1記載の発明では、複数のグリフデータを有し、該各グリフデータは、輪郭点の集合であるアウトライン情報と該アウトライン情報を制御するヒンティング情報によって構成されてなるアウトラインフォントファイルにおける輪郭線のヒンティング処理方法において、輪郭線を構成する所定の輪郭点に対しては、線幅補正、線消え防止およびかすれ防止を含む第1のヒンティング処理を施し、それ以外の輪郭点に対しては、該第1のヒンティング処理された点との相対位置を保つように移動する第2のヒンティング処理を施すことを特徴としている。

【0011】請求項2記載の発明では、曲線を制御する始点、終点を除く点に対して、前記第2のヒンティング処理を施すことを特徴としている。

【0012】請求項3記載の発明では、縦横ステムを構成する輪郭点に対して前記第1のヒンティング処理を施し、それ以外の輪郭点に対して前記第2のヒンティング処理を施すことを特徴としている。

【0013】請求項4記載の発明では、縦横ステムを構成する輪郭点と曲線の始点と終点に対して前記第1のヒンティング処理を施すことを特徴としている。

【0014】請求項5記載の発明では、曲線中の方向変化を生じる部分に対して、前記第1のヒンティング処理を施すことを特徴とする請求項4記載のヒンティング処理方法。

【0015】請求項6記載の発明では、長方形の対角線上の2点に対して前記第1のヒンティング処理を施し、他の2点に対して前記第2のヒンティング処理を施すことを特徴としている。

【0016】請求項7記載の発明では、グリフにおけるウロコの先端点に対して前記第2のヒンティング処理を施すことを特徴としている。

【0017】請求項8記載の発明では、本来等しい幅であるべきステムが変倍後においてもその線幅を等しく保つように線幅を補正するヒンティング処理方法において、該線幅と変倍後のサイズは所定の関数関係にあることを特徴としている。

【0018】請求項9記載の発明では、変倍後のサイズと該サイズに応じた線幅との対応を記憶したテーブルを参照して線幅を補正することを特徴としている。

【0019】請求項10記載の発明では、変倍後のサイズに応じて、請求項8記載の関数または請求項9記載のテーブルの何れかを用いて線幅を補正することを特徴としている。

【0020】請求項11記載の発明では、前記請求項1ないし7記載の処理と前記請求項8ないし10記載の線幅補正処理を組み合わせることを特徴としている。

【0021】

【作用】輪郭線の始点と終点の点番号をNs，Neに保持し、NsとNeのx座標の差をX'に、y座標の差をY'に保持する。X（N）を点Nのx座標、Y（N）を点Nのy座標とし、X'（N）を点Nの移動後のx座標、Y'（N）を点Nの移動後のy座標とし、始点の次の点をNに設定し、点Nが終点Nsになるまで、X'（N）とY'（N）を計算し、NをN＋1にして繰り返す。これにより、点Nsと点Neに挟まれた点が、最初の相対位置を保つように移動処理され、ヒンティング処理後のグリフの形状を美しくすることができる。

【0022】

【実施例】以下、本発明の一実施例を図面を用いて具体的に説明する。本実施例においては、各輪郭点は、そのグリフ内で一意に定まる連続した番号で参照され、図9に示すようにその番号は輪郭線をたどる順に1ずつ増えるものとする。また、曲線の表現としては二次スプライン（これはいわゆるコニック曲線である）を用いて説明するが、本実施例はこれに限定されるものではなく、他の曲線の表現であってもよい。

【0023】また、本実施例の前提となるアウトラインフォントファイル構成は、図15に示すように、n個のグリフデータ1（グリフ1，グリフ2，・・・グリフn）と共有データ2によって構成されている。

【0024】また、図16に示すようにグリフデータは、アウトライン情報11とヒンティング情報12からなっている。そして、アウトライン情報11は、連続的に番号を付けられた輪郭点とその座標データからなり、ヒンティング情報12は、アウトライン情報11を制御する命令列（例えば、点1を右に1ドットシフトするという命令など）から構成されている。図17は、共有データ2の中に輪郭点を制御するためのサブルーチン群（サブルーチン1からサブルーチンm）が含まれている図である。

【0025】なお、本実施例で用いる従来のヒンティング処理について簡単に説明する。

（1）線幅補正

変倍されたフォントのアウトラインをそのままラスタライズすると、その置かれた位置によっては、本来同じ幅にデザインされていても、生成されるドットイメージとしては異なる幅になってしまうことがある。これを補正処理するのがいわゆる線幅補正である。これは以下のようにして処理する。

【0026】1．縦ステムの左側の縦線、横ステムの下側の横線をピクセル境界やピクセル中心などの特定の位置に移動し、そこから特定の距離の位置に対応する線を移動する。

2．縦横ステムの中心線をピクセル境界やピクセル中心などの特定の位置に移動し、そこから特定の距離の位置

に輪郭線を移動する。

3．各文字に複数の特徴点を設定し、各ステムの位置をその特徴点からの距離として管理する。この制御点をピクセル境界やピクセル中心などの特定の位置に移動し、そこから特定の距離の位置に各ステムを移動する。

【0027】（2）線消えの防止

変倍によってそのサイズが小さくなると、あるステムがピクセルとピクセルの間に入ってしまうことがある。そのためにそのステムはドットとしては現われない。これを防止するために、次の方法が用いられる。

【0028】1．アウトラインのそのステムに関わる部分のみを移動してドットとして現われるようにする。

2．ステムの最小幅を例えば1に規定しておくことにより、必ずそのステムが現われるようにする。

3．ラスタライザの機能として、このような線消えが起きないようにする。このような処理は、斜線や曲線によって構成される縦横に尖った部分についても適用される。

【0029】（3）かすれの防止

変倍によってそのサイズが小さくなると、斜線や曲線においてその一部がピクセルとピクセルの間に入ってしまうことがある。その結果、かすれて見えることになり、美しさが損なわれる。これを防止するために、次の方法が用いられる。

【0030】1．斜線や曲線の最小幅を例えば1に規定しておくことにより、必ずその斜線や曲線が現われるようにする。

2．ラスタライザの機能として、このようなかすれが起きないようにする。

【0031】（実施例1）本発明は、上記ヒンティング情報12における輪郭点の制御方法に係り、図1は、本実施例に係る相対位置を保存して移動処理するヒンティング処理のフローチャートである。図1において、まず輪郭線の始点と終点の点番号をNs，Neにそれぞれ保持する（ステップ101）。NsとNeのx座標の差をXに、y座標の差をYに保持する（ステップ102）。このときXまたはYが0であれば例外として処理する（ステップ108）。

【0032】X（N）を点Nのx座標、Y（N）を点Nのy座標とし、X’（N）を点Nの移動後のx座標、Y’（N）を点Nの移動後のy座標とする。始点の次の点をNに設定する（ステップ104）。そして、点Nが終点Nsになるまで、

X’（N）＝｛（X（N）－X（Ns））＊（X’（Ne）－X’（Ns））／X｝＋X’（Ns）

Y’（N）＝｛（Y（N）－Y（Ns））＊（Y’（Ne）－Y’（Ns））／Y｝＋Y’（Ns）

を計算し、Nを N＋1にして繰り返す（ステップ105，106，107）。

【0033】上記移動後の位置の計算は、比例配分の方法であり、これにより、点Nsと点Neに挟まれた点を、最初の相対位置を保つように移動することができる。本実施例の処理を用いて実際に図形を移動する例を図2を用いて説明する。なお、ここでは元の図形をあるサイズに縮小してあるものとして説明する。

【0034】図2（a）は、初期状態を示す図であり、点aで始まり点bをオフポイント（曲線の始点と終点にはさまれた、曲線を制御するための点をいう）として点cに至る曲線と、点cに始まり点dをオフポイントとして点eに至る曲線が滑らかに接続している状態を表している。各点の相対位置は、

X（b）－X（a）：X（c）－X（b）：X（d）－X（c）：X（e）－X（d）＝2：5：5：4

Y（b）－Y（a）：Y（c）－Y（b）：Y（d）－Y（c）：Y（e）－Y（d）＝7：3：3：－2

である。

【0035】図2（b）は、従来のヒンティング処理によって、これらの輪郭点が整数座標上に丸められた図を示し、この図から明らかなように、曲線の接続が正しく保存されていない。

【0036】そこで、本実施例では、点aと点eのみに従来のヒンティング処理を施し、残る点に対して、前述した相対位置を保つように点を移動することによって、図2（c）に示す曲線が得られる。図2（c）の曲線から明らかなように、曲線の接続が正しく保存されていることが分かる。このように、実施例1では、デザイン時の輪郭点の相対的な位置が保存されるので、ヒンティング処理後もグリフの形状が極端に歪むことが防止され、特に連続した曲線においてその効果が顕著に現われる。また、ウロコやトメといったグリフの特徴的な形状を統一することができる。

【0037】（実施例2）漢字のように画数の多い文字を表示する際に重要なことは、ウロコやトメの如き特徴を少なくともそのグリフ内で統一して表示することである。図4は、グリフの一例を示す図で、点nと点iの部分の形状がグリフ内において統一されている必要がある。勿論、フォント全体を通じて形状が統一されていることが望ましいが、そのために相当な処理を要する。そこで、本実施例では、一つのグリフ中においては同じ形状になるように処理するものである。

【0038】図4のグリフにおいて、点oと点j、点mと点hに対して、それぞれ同じヒンティング処理が施される。従って、本実施例では、それらの点との相対位置を保つことによって、点nと点iの部分が同一の形状のドットイメージを生成するようにしたものである。

【0039】図3は、実施例2の処理フローチャートである。なお、ここでは説明を簡単にするために、輪郭線の始点はオフポイントではないものとする。まず、輪郭線の始点をPnとする（ステップ201）。Pnがオフポイントでなければ（ステップ204）、通常のヒンテ

ィング処理を施す（ステップ２０５）。その後ＰｎをＰｎ＋１にする（ステップ２０６）。この処理をＰｎが輪郭線の終点を越えるまで繰り返す（ステップ２０３）。

【００４０】再び、輪郭線の始点をＰｎとし、その直前の点（この時点では輪郭線の終点）をＰｐとする（ステップ２０７）。Ｐｎがオフポイントでなければ何もしない。Ｐｎがオフポイントである間、ＰｎをＰｎ＋１にすることを繰り返す（ステップ２１０，２１１，２１２）。このとき、輪郭線の終点に至った場合には始点に戻る（ステップ２１５）。Ｐｎがオフポイントでなくなった時点で、ＰｐとＰｎで挟まれた点に対して図１に示した相対位置を保つ移動処理を施す（ステップ２１３）。ＰｎをＰｎ＋１に、Ｐｐをその直前の点にする（ステップ２１４）。以上の処理をＰｎが輪郭線の終点を越えるまで繰り返す。

【００４１】このように、実施例２によれば、トメなどの一つの曲線で表される形状については、グリフ内でその形状を統一することができる。

【００４２】〈実施例３〉明朝体のように縦横ステムに特徴のあるフォントにおいては、それらに対するヒンティング処理が重要となる。つまり、逆にいえばそれ以外の点については、ヒンティング処理はさほど重要ではないことになる。

【００４３】本実施例では、図４に示すグリフにおいて、縦横ステムを構成する点ａ，ｂ，ｃ，ｇ，ｈ，ｊ，ｋ，ｌ，ｍ、ｏに対して従来のヒンティング処理（例えば、線幅の補正処理など）を施し、それ以外の点ｄ，ｅ，ｆ，ｉ，ｎに対しては、本実施例の図１の相対位置を保つ移動処理を施すだけで、十分に美しい形状を得られるようにしたものである。

【００４４】図５，図６は、実施例３の処理フローチャートである。図５，図６において、輪郭線の始点をＰｓ、Ｐｓの次の点をＰｅとする（ステップ３０１）。ＰｓとＰｅのｘ座標が同じであるならば、ＰｓからＰｅに至る線分は垂直線である（ステップ３０２）。そして、その線分は、縦ステムの左右端の何れかであるが（ステップ３０７）、もし、Ｐｓのｙ座標がＰｅのそれよりも大きければ右端、そうでなければ左端と判定され、これらに対しては縦ステムの幅を所定幅にするようなヒンティング処理を施す（ステップ３０８，３０９）。

【００４５】ＰｓとＰｅのｘ座標が異なり、ｙ座標が等しい場合も（ステップ３０３）全く同様に横ステムの上端であるか下端であるかを判定する（ステップ３０４）。そして、これらに対しては横ステムの幅を所定幅にするようなヒンティング処理を施す（ステップ３０５，３０６）。

【００４６】ＰｓからＰｅに至る線分が、上記いずれのステムでもなければ何もしない。Ｐｓが輪郭線の終点になるまで、つまり輪郭線上のすべての線分について以上の処理を行う（ステップ３１０，３１１，３１２，３１３）。

【００４７】再び、輪郭線の始点をＰｓとして、Ｐｓの次の点をＰｅとする（ステップ３１４）。ＰｓからＰｅに至る線分が水平または垂直であるならば、何もしない（ステップ３２４以下）。そうでなければ、その始点ＰｓをＰｃに保持する（ステップ３１６）。ＰｓをＰｅとし、Ｐｓの次の点をＰｅとする（ステップ３１７，３１８，３２０）。ＰｓからＰｅに至る線分が水平または垂直でない間、この処理を繰り返す（ステップ３２２）。

【００４８】ＰｓからＰｅに至る線分が水平または垂直になったら、ＰｃとＰｓに挟まれた一連の点に対して、図１の相対位置を保つ移動処理を施す（ステップ３２２，３２３）。なお、縁返し中に輪郭線の終点が来たときはエラーであるが、ここで図１の相対位置を保つ移動処理を施して処理を終了する（ステップ３２１）。その後、ＰｓをＰｅとし、Ｐｓの次の点をＰｅとする（ステップ３２７）。そして、ステップ３１５に戻る。このときＰｓが輪郭線の終点に来れば処理を終了する（ステップ３２５）。

【００４９】本実施例によれば、縦横ステムに従来のヒンティング処理を施すだけで、グリフ全体として整った形状にすることができ、また縦横ステム以外の部分には、移動処理のヒンティング処理を施しているので、その処理データ量を大幅に削減することが可能となる。

【００５０】〈実施例４〉図９（ａ）のグリフに対して、上記した実施例３の処理を適用した場合、点０，１，２の如きステム上の点に対しては従来のヒンティング処理が施される。これに対して、点１３から点１９に至る曲線群や点２３から点２７に至る曲線群は、その縦横ステムのヒンティング処理に伴って図１の相対位置を保つ移動処理が施される。このため、ステムに対する従来のヒンティング処理によってステムの太さに変化が生じたとき、曲線部の太さとの統一性が失われる可能性がある。

【００５１】そこで、本実施例では、点１３，１９や点２３，２７を上下に移動するようなヒンティング処理を施し、その後、図１の相対位置を保つ移動処理を施すことによって、曲線部を滑らかにしつつ全体の太さを調整するようにしたものである。

【００５２】図７，図８は、実施例４の処理フローチャートである。この処理は、基本的には図５，図６の処理と同じものである。ただし、ここでは縦横ステムに対する従来のヒンティング処理（図５のステップ３０５，３０６，３０８，３０９）を「補正処理（ステップ４０３）」としてまとめている。

【００５３】まず、Ｐｓを輪郭線の始点とし、ＰｅをＰｓの次の点とする（ステップ４０１）。ＰｓからＰｅに至る直線が縦横ステムを構成するならば、そのためのヒンティング処理を施す（ステップ４０３）。そうでなければ、ＰｓからＰｅに至る直線が縦横ステムとなるま

で、逐次ＰｓとＰｅを増やしながら輪郭線をたどる（ステップステップ４０７、ステップ４１３までの処理）。

【００５４】輪郭線の終点に至ったところで（ステップ４０５、ステップ４１１）、再びＰｓを輪郭線の始点とし、ＰｅをＰｓの次の点とする（ステップ４１４）。Ｐｓが曲線部の開始点であれば（ステップ４１５）、ＰｃにＰｓを保持する（ステップ４２０）。Ｐｅが曲線部の終点になるまでＰｓとＰｅを増やしながら輪郭線をたどる（ステップ４２１以下を繰り返す）。曲線部の始点、終点であるＰｃとＰｅに必要なヒンティング処理を施す。その後、これらＰｃとＰｅに挟まれた点に対して、図１の移動処理を施す（ステップ４２６）。Ｐｓが曲線部の終点でない間、ＰｓとＰｃを増やしながら輪郭線をたどり、以上の処理を繰り返す。

【００５５】本実施例によれば、縦横ステムと曲線（または曲線群）の始点、終点だけに従来のヒンティング処理を施すことにより、美しいヒンティング処理結果を得ることができる。特に、本実施例の簡単な移動処理によって曲線部全体にわたってヒンティング処理が施されることになる。

【００５６】〈実施例５〉本実施例は、方向変化を生じる部分に対してヒンティング処理を施すようにしたものである。図９（ｂ）は、漢字の一部を示し、点５から点６に至る直線と点１６から点１７に至る直線は、横ステムを構成している。このような図に対して、従来は前述した技術を斜み合わせてヒンティング処理を施していう。すなわち、点５から点６に至る直線と、点１６から点１７に至る直線によって構成される横ステムに対しては、線幅補正と線消え対策が施され、また点８から点１０に至る曲線や点１２から点１４に至る曲線に対しては、一種の線消え対策が施されている。

【００５７】ところが、これらの処理を独立に行うと、曲線の接続が滑らかでなくなる恐れがあり、また全体の形状がゆがんでしまいやすい。そこで、上記実施例４で説明した方式に従って、曲線部に対して移動処理を施すことを考える。点６と点１６に対しては従来のヒンティング処理が施される。この処理の結果、これらの点は上下に移動することになる。

【００５８】そして、点６と点１６の間の曲線に対して、移動処理を施すと、点７から点１５までの点は、その相対位置を保つように移動される。その結果、これらの曲線群は滑らかにつながったまま、その位置を変えることができる。

【００５９】ところが、点６と点１６は、上下方向にしか移動しないため、この移動処理によっては例えば、点８から点１０に至る曲線に一種の線消え対策が施されていない。そのため点８から点１０に至る曲線部分がドットとしては汲示されなくなってしまう可能性がある。点１２から点１４に至る曲線についても同様である。

【００６０】従って、このような場合には、点８から点１０に至る曲線と点１２から点１４に至る曲線に対しては、従来のヒンティング処理を施した後、移動処理を実行すればよいことになる。すなわち、曲線や斜線が、その方向を大きく変えるときに、従来のヒンティング処理を施せばよい。

【００６１】本実施例では、点をたどる方向の変化が９０度を下回る（急な角をなす）ものについて従来のヒンティング処理を適用する例について示した。しかし、どのような条件のときにこれを適用するかについては種々の方法があり、例えば、９０度ではなく、１２０度を下回るときに適用するようにしたり、あるいはまた急な傾きの部分からなだらかな傾きに変わる部分や、その逆の部分にこれを適用することもできる。

【００６２】本実施例によれば、縦横ステムと曲線（または曲線群）の始点、終点と方向の変化点にヒンティング処理を施すことにより、美しいヒンティング処理結果を得ることができる。特に、簡単な処理によって曲線部全体にわたってヒンティング処理が施されることになる。

【００６３】〈実施例６〉図１０に示す、長方形の輪郭線に対してヒンティング処理を施すことを考える。例えば、点ａを基準にしてヒンティング処理をする。点ｂは、その縦方向には点ａと同じ処理を行い、横方向には縦ステムの線幅を補正するための処理をする。点ｄは、横方向には点ａと同じ処理を行い、縦方向には撲ステムの線幅を補正するための処理をする。

【００６４】残った点ｃは、縦横方向ともに線幅補正のための処理を行わなければならず、処理に相当の時間を要する。ところが、前述した本実施例の相対位置を保存する移動処理を用いることによって、この点に対するヒンティング処理を不要にすることができる。つまり、本実施例の相対位置を保存する移動処理は、元もと等しいｘ，ｙ座標は移動処理後も等しいことを保証するからである。同様の理由から、点ａについてもヒンティング処理が不要になり、結局長方形に関しては対角線上の２点に対してのみヒンティング処理を施せば十分である。

【００６５】本実施例によれば、従来４つの輪郭点すべてに施していたヒンティング処理が、対角線上の２点だけにヒンティング処理を施せば済み、処理時間、データ量ともに格段に削減することができる。

【００６６】〈実施例７〉本実施例は、ウロコの頂点に対して、相対位置を保存する移動処理を施す実施例である。すなわち、図１１は、あるグリフのウロコ部分に対するヒンティング処理の前後の図であり、（ａ）は、変倍された原図形、（ｂ）は、各輪郭点を整数座標上に四捨五入した従来のヒンティング処理後の図である。従来から行われてきたヒンティング処理は、そのほとんどが切り捨て、切り上げなどの方法を採っているので、ヒンティング処理後の輪郭線の形状は、その置かれた位置によって全く異なったものとなり、当然ラスタライズされ

た結果も全く異なったものになる。

【0067】そこで、本実施例では、ウロコの頂点にのみ、相対位置を保存する移動処理を施すことにより、図11（c）に示すように、ウロコの形状を統一することができる。このように、本実施例によれば、簡単な処理によって、一つのグリフ内でウロコの形状を統一することが可能になる。

【0068】〈実施例8〉従来、線幅補正を行う際の線幅は、一定のサイズでデザインされたものを、その表示サイズにしたがって変倍することによって得ていた。つまり、その線幅の変化は、図12に示すように線形に変化していた。このため、従来技術で説明したように、小サイズでは線が太く、大サイズでは線が細くなるという印象を与えることになる。また、サイズが十分小さくなると、ステムが消えてしまうことにもなる。

【0069】そこで、本実施例では、図13に示すような各種の関数を用いて線幅を決定するものである。すなわち、図13（a）は、極小サイズではある最小幅（例えば、1ドット）になるようにし、それ以上のサイズでは従来と同様に線形に変化する関数にしたものである。（b）は、例えば2次曲線のような、小サイズでは変化が小さく、大サイズでは変化が大きくなる関数の例である。（c）は、（a）と（b）をつなげた関数であり、すなわち、極小サイズではある最小幅（例えば、1ドット）になり、ある大サイズまでは2次曲線のように変化し、それより大きなサイズでは直線的に変化する関数である。なお、本実施例は特定の関数に限定されるものではなく、各種の関数が適用可能である。

【0070】本実施例によれば、少ない記憶容量で、種々の特性の線幅処理を行うことができる。

【0071】〈実施例9〉本実施例は、上記関数をテーブルを用いて実現した実施例である。図14（a）は、サイズと線幅とが1対1の対応関係にあるテーブルの構成例を示し、（b）は、ある範囲のサイズを与えたときの対応する線幅を生成する多対1の対応関係にあるテーブルの構成例を示す図である。なお、これらの数値は、単なる例示にすぎない。

【0072】本実施例によれば、テーブルを用いることにより、記憶容量は増えるものの、関数計算に比べて高速の処理が可能になる。

【0073】〈実施例10〉本実施例は、上記した実施例8と9を組み合わせた実施例である。すなわち、例えば30～50ドットという、よく使用されるサイズ、あるいは20～30ドットという、きめ細かな処理が必要となるサイズに対してはテーブルを用いて線幅計算し、その他のサイズに対しては関数を用いる。これによって、必要なサイズにおいては処理速度が速く、使用頻度の低いサイズにおいては記憶容量が少なくて済む。

【0074】〈実施例11〉上述した本実施例1から7は、相対位置を保存する移動処理の用い方に関する実施例であり、特に実施例1は一般的なもの、実施例2から5は曲線やステムに関するもの、実施例6は長方形の処理、実施例7はウロコの処理に関するものである。従って、これらの処理の内の幾つかを組み合わせて用いることが可能となる。さらに、実施例8から11に示した線幅の決定方法も同時に適用することにより、更に美しいフォントを得ることができる。このように、本実施例では、前述した各実施例の全ての長所を選択的に使用することができる。

【0075】本発明は、一般のアウトラインフォントファイルのヒンティング処理方法について開示した。しかし、この方法は、True Typeのようにそれ自身にプログラムとしてヒンティング情報が含まれているシステムで用いれば、より一層効果的である。特に、True Typeではフォントファイル内で共有されるデータ群を有し、その中には各グリフのヒンティング手順から呼びだされるサブルーチン群を含むように構成されているので、上記した本実施例の処理方法の共通部分や繰返し使用される部分などをサブルーチンとして登録することにより、一層のデータ量の削減が可能になる。

【0076】

【発明の効果】以上、説明したように、請求項1記載の発明によれば、ヒンティング処理後においてもデザイン時の輪郭点の相対的な位置が保存されるので、グリフの形状が極端に歪むことがなくなる。

【0077】請求項2記載の発明によれば、グリフ内でトメなどの形状を統一することができる。

【0078】請求項3記載の発明によれば、縦横ステムに第1のヒンティング処理を施すだけでグリフ全体として整った形状にすることができ、またその処理量を大幅に削減することが可能となる。

【0079】請求項4記載の発明によれば、縦横ステムと曲線の始点、終点に第1のヒンティング処理を施しているので、美しいヒンティング処理結果が得られる。

【0080】請求項5記載の発明によれば、縦横ステムと曲線の始点、終点と方向の変化点に対して第1のヒンティング処理を施しているので、美しいヒンティング処理結果が得られる。

【0081】請求項6記載の発明によれば、長方形の対角線上の2点だけに第1のヒンティング処理を施せば済むので、処理時間、データ量ともに大幅に削減することができる。

【0082】請求項7記載の発明によれば、一つのグリフ内でウロコの形状を統一することが可能になる。

【0083】請求項8記載の発明によれば、少ない記憶容量で、種々の特性の線幅処理を行うことができ、サイズに適応した美しいフォントを得ることができる。

【0084】請求項9記載の発明によれば、テーブルを参照することにより高速の処理が可能になる。

【0085】請求項10記載の発明によれば、必要なサ

イズにおいては処理速度が速く、使用頻度の低いサイズにおいては記憶容量が少なくて済むとともに、サイズに適応して美しく処理することができる。

【００８６】請求項１１記載の発明によれば、全ての長所を選択的に使用することができるので、更に美しいフォントを得ることができる。

【図面の簡単な説明】

【図１】本実施例に係る相対位置を保存して移動処理するヒンティング処理のフローチャートである。

【図２】（ａ）は、初期状態の曲線を示す図であり、（ｂ）は、従来のヒンティング処理によって輪郭点が整数座標上に丸められた図であり、（ｃ）は、本実施例によって処理された曲線を示す図である。

【図３】実施例２の処理フローチャートである。

【図４】グリフの一例を示す図である。

【図５】実施例３の処理フローチャートである。

【図６】図５の処理フローチャートの続きである。

【図７】実施例４の処理フローチャートである。

【図８】図７の処理フローチャートの続きである。

【図９】（ａ）、（ｂ）は、本発明が適用されるグリフの他の例である。

【図１０】長方形の輪郭線に対して本発明が適用される＊例である。

【図１１】ウロコ部分に対するヒンティング処理の前後の図であり、（ａ）は、変倍された原図形、（ｂ）は、各輪郭点を整数座標上に四捨五入した従来のヒンティング処理後の図、（ｃ）は、本実施例のウロコの頂点にのみ相対位置を保存する移動処理を施した図である。

【図１２】従来のサイズと線幅の変化を示す図である。

【図１３】（ａ），（ｂ），（ｃ）は、本実施例の線幅を決定する各種の関数である。

【図１４】（ａ）、（ｂ）は、線幅を生成するテーブルの構成を示す図である。

【図１５】アウトラインフォントファイルの構成を示す図である。

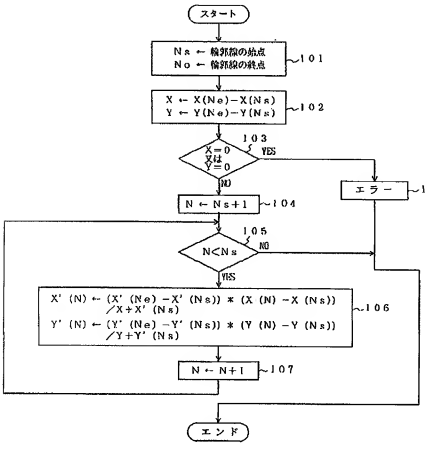
【図１６】グリフデータの構成を示す図である。

【図１７】共有データの中に輪郭点を削御するためのサブルーチンが含まれている図である。

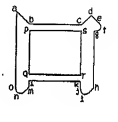
【符号の説明】

１　グリフデータ
２　共有データ
１１　アウトライン情報
１２　ヒンティング情報

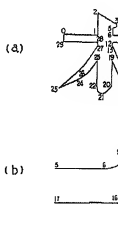
【図１】



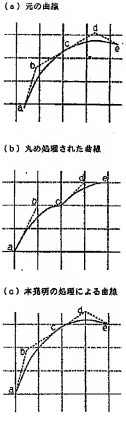
【図４】



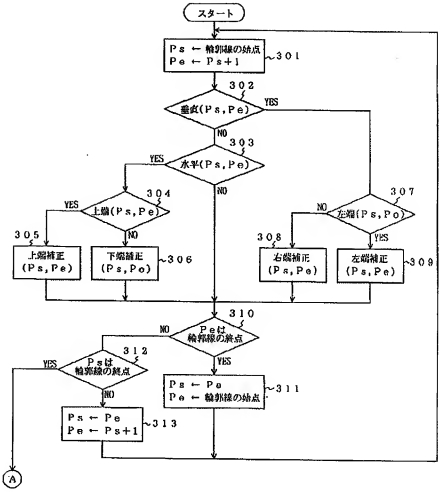
【図９】



【図２】

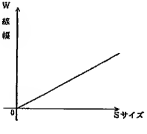


【図５】



【図１０】

【図１２】



【図１４】

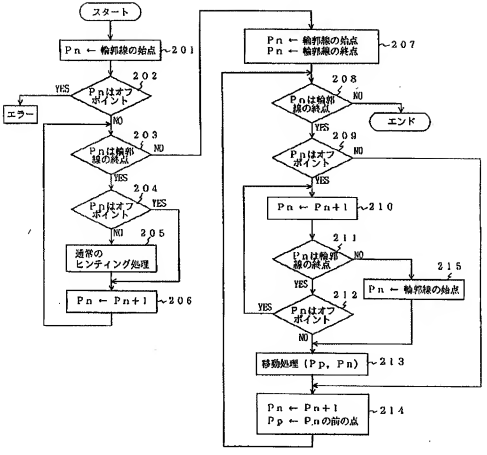
(a)

サイズ	線幅
1	1
2	1
3	1
4	2
5	2
・	・
・	・
・	・

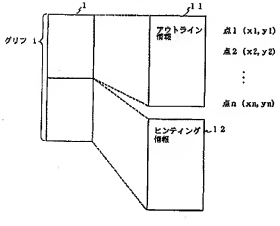
(b)

サイズ		線幅
始点	終点	
1	3	1
4	7	2
8	16	3
17	26	4
・	・	・
・	・	・
・	・	・

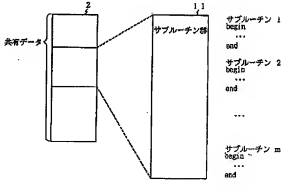
【図３】


スタート
Ｐｎ ← 輪郭線の始点
201
202
Ｐｎはオフポイント
YES
エラー
NO
203
Ｐｎは輪郭線の終点
NO
YES
204
Ｐｎはオフポイント
YES
NO
205
通常のヒンティング処理
Ｐｎ ← Ｐｎ＋１
206
Ｐｎ ← 輪郭線の始点
Ｐｎ ← 輪郭線の終点
207
208
Ｐｎは輪郭線の終点
NO
エンド
YES
209
Ｐｎはオフポイント
NO
YES
Ｐｎ ← Ｐｎ＋１
210
211
Ｐｎは輪郭線の終点
NO
215
Ｐｎ ← 輪郭線の始点
YES
212
Ｐｎはオフポイント
YES
NO
移動処理（Ｐｐ，Ｐｎ）
213
Ｐｎ ← Ｐｎ＋１
Ｐｐ ← Ｐｎの前の点
214

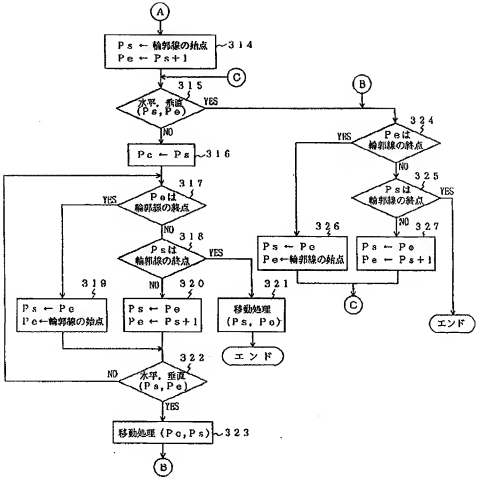
【図１６】


1
11
グリフ 1
アウトライン情報
点1 (x1,y1)
点2 (x2,y2)
点n (xn,yn)
ヒンティング情報
12

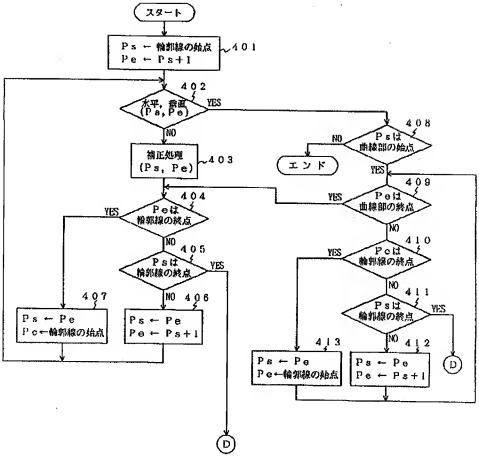
【図１７】


2
11
共有データ
サブルーチン群
サブルーチン 1
begin
end
サブルーチン 2
begin
end
サブルーチン m
begin
end

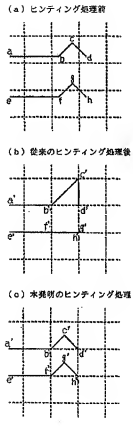
【図6】


A
Ps ← 輪郭線の始点
Pe ← Ps+1
314
C
315
水平，垂直
(Ps,Pe)
YES
NO
B
324
Peは
輪郭線の終点
YES
NO
325
Psは
輪郭線の終点
YES
NO
326
Ps ← Pe
Pe←輪郭線の始点
327
Ps ← Pe
Pe ← Ps+1
C
エンド
Pc ← Ps
316
317
Peは
輪郭線の終点
YES
NO
318
Psは
輪郭線の終点
YES
NO
319
Ps ← Pe
Pe←輪郭線の始点
320
Ps ← Pe
Pe ← Ps+1
321
移動処理
(Ps, Pe)
エンド
322
水平，垂直
(Ps,Pe)
NO
YES
移動処理（Pc,Ps）
323
B

【図７】


スタート
Ps ← 輪郭線の始点
Pe ← Ps＋1
401
402
水平、垂直
(Ps, Pe)
YES
NO
補正処理
(Ps, Pe)
403
404
Peは
輪郭線の終点
YES
NO
405
Psは
輪郭線の終点
YES
NO
406
Ps ← Pe
Pe ← Ps＋1
407
Ps ← Pe
Pe←輪郭線の始点
D
408
Psは
曲線部の始点
NO
YES
エンド
409
Peは
曲線部の終点
YES
NO
410
Peは
輪郭線の終点
YES
NO
411
Psは
輪郭線の終点
YES
NO
412
Ps ← Pe
Pe ← Ps＋1
413
Ps ← Pe
Pe←輪郭線の始点
D

【図１１】



【図１３】